In this post, I'll share 19 powerful JavaScript one-liners you'll want to bookmark.

> ***Before we get started, don't forget to subscribe to my newsletter!***
> *Get the latest tips, tools, and resources to level up your web development skills delivered straight to your inbox. Subscribe here!*

Now, let's jump right into it!

1. Capitalize Text

```
const capitalize = str => str.charAt(0).toUpperCase() + str.slice(1);

// Example:
console.log(capitalize("hello")); // Output: Hello
```

This is useful for formatting names, headings, or labels.

2. Compare Arrays and Objects for Equality

```
const isEqual = (a, b) => JSON.stringify(a) === JSON.stringify(b);

// Examples:
console.log(isEqual([1, 2, 3], [1, 2, 3])); // Output: true
console.log(isEqual([1, 2], [2, 1])); // Output: false — different order
console.log(isEqual({ a: 1, b: 2 }, { a: 1, b: 2 })); // Output: true — identical
objects
console.log(isEqual({ a: 1, b: 2 }, { b: 2, a: 1 })); // Output: false — key order
matters in JSON.stringify
console.log(isEqual({ a: 1, b: 2 }, { a: 1 })); // Output: false — missing
property
console.log(isEqual(null, null)); // Output: true — both null
console.log(isEqual(undefined, undefined)); // Output: false —
JSON.stringify(undefined) is undefined
console.log(isEqual(42, 42)); // Output: true — primitives equal
console.log(isEqual(42, "42")); // Output: false — different types
console.log(isEqual([1, 2, { x: 3 }], [1, 2, { x: 3 }])); // Output: true — nested
object
```

This is useful for detecting state changes or deep comparisons.

3. Check If an Array Is Empty

```
const isEmpty = arr => Array.isArray(arr) && arr.length === 0;

// Examples:
console.log(isEmpty([])); // Output: true (empty array)
console.log(isEmpty([1, 2, 3])); // Output: false (non-empty array)
console.log(isEmpty('')); // Output: false (string, not an array)
console.log(isEmpty(null)); // Output: false (null)
console.log(isEmpty({})); // Output: false (object, not an array)
console.log(isEmpty(undefined)); // Output: false (undefined)
console.log(isEmpty([undefined])); // Output: false (array with one element)
console.log(isEmpty([null])); // Output: false (array with one element)
```

This is useful for showing fallback UIs when there's no data.

4. Check If a Number Is Even

```
const isEven = n => !(n % 2);

// Example:
console.log(isEven(7)); // Output: false
console.log(isEven(2)); // Output: true
console.log(isEven(0)); // Output: true
console.log(isEven(-4)); // Output: true
console.log(isEven(-3)); // Output: false
console.log(isEven(1.5)); // Output: false
console.log(isEven("4")); // Output: true (string coerced to number 4)
console.log(isEven("abc")); // Output: true (NaN % 2 = NaN, !NaN is true —
unexpected!)
```

This is useful for conditional logic, games, and filters.

5. Copy to Clipboard

```
const copy = text => navigator.clipboard.writeText(text);
```

This is useful for copying code, URLs, or promo codes with a click.

6. Detect Dark Mode

```
const isDark = window.matchMedia('(prefers-color-scheme: dark)').matches;

// Example:
console.log(isDark); // Output: true
```

This returns true if the user prefers dark mode, and useful for auto-switch themes based on OS settings.

7. Get Day Name from Date

```
const getDay = date => new Date(date).toLocaleDateString('en-US', { weekday:
'long' });

// Examples:
console.log(getDay('2025-07-26')); // Output: Saturday
console.log(getDay('2025-07-27')); // Output: Sunday
console.log(getDay(new Date())); // Output: Current day name, e.g., "Saturday"
console.log(getDay('invalid-date')); // Output: Invalid Date
```

This is useful for calendars, task reminders, and scheduling apps.

8. Get a Random Element

```
const randomItem = arr => arr[Math.floor(Math.random() * arr.length)];

// Example:
const colors = ['red', 'green', 'blue', 'yellow'];
console.log(randomItem(colors)); // Output: yellow
console.log(randomItem(colors)); // Output: red
console.log(randomItem(colors)); // Output: green
```

This picks a random item from an array, and this is useful for random quotes, colors, tips, or products.

9. Get Query Parameters as Object

```
const getParams = url => Object.fromEntries(new URLSearchParams(new
URL(url).search).entries());

// Examples:

// Basic URL with parameters
console.log(getParams('https://example.com?name=Shefali&lang=js')); // Output: {
name: "Shefali", lang: "js" }

// URL with empty parameters
console.log(getParams('https://example.com?foo=&bar=123')); // Output: { foo: "",
bar: "123" }

// URL with no parameters
console.log(getParams('https://example.com')); // Output: { }

// URL with encoded characters
console.log(getParams('https://example.com?search=hello%20world')); // Output: {
search: "hello world" }
```

This is useful for reading filters, UTM tags, or search queries.

10. Insert Element at a Specific Position

```
const insertAt = (arr, val, i) => [...arr.slice(0, i), val, ...arr.slice(i)];

// Examples:
console.log(insertAt([1, 2, 3], 99, 1)); // Output: [1, 99, 2, 3]
console.log(insertAt(['a', 'b', 'c'], 'x', 0)); // Output: ['x', 'a', 'b', 'c']
console.log(insertAt([true, false], true, 2)); // Output: [true, false, true]
console.log(insertAt([], 'start', 0)); // Output: ['start']
```

This will insert val into arr at index i and useful for reordering lists, inserting dynamic content.

11. Merge Two Arrays

```
const merge = (a, b) => [...a, ...b];

// Examples:
console.log(merge([1, 2], [3, 4]));           // Output: [1, 2, 3, 4]
console.log(merge([], [1, 2]));               // Output: [1, 2]
console.log(merge([1, 2], []));               // Output: [1, 2]
console.log(merge([], []));                   // Output: []
console.log(merge(['a'], ['b', 'c']));        // Output: ['a', 'b', 'c']
console.log(merge([true, false], [false]));   // Output: [true, false, false]
```

This is useful for merging fetched results or local + remote data.

12. Generate a Random Hex Color

```
const randomColor = () => '#' + Math.floor(Math.random() * 16777215).toString(16);

// Examples:
console.log(randomColor()); // Output: #b35d91
console.log(randomColor()); // Output: #94a444
console.log(randomColor()); // Output: #16b3ce
```

This is useful for creating random backgrounds, themes, and colorful elements.

13. Remove Duplicates from an Array

```
const unique = arr => [...new Set(arr)];

// Examples:
console.log(unique([1, 2, 2, 3, 4, 4])); // Output: [1, 2, 3, 4]
console.log(unique(['a', 'b', 'a', 'c'])); // Output: ['a', 'b', 'c']
console.log(unique([true, false, true])); // Output: [true, false]
console.log(unique([1, '1', 1])); // Output: [1, '1'] (different types)
console.log(unique([])); // Output: []
```

This is useful for tags, filters, and user input cleanup.

14. Reverse a String

```
const reverse = str => str.split('').reverse().join('');

// Example:
console.log(reverse("hello")); // Output: olleh
```

This is useful for fun puzzles, palindrome checks, and string transforms.

15. Sleep / Delay Execution

```
const sleep = ms => new Promise(res => setTimeout(res, ms));
```

This will wait for a given number of milliseconds and is useful for simulating loading, pausing between async steps.

16. Shuffle an Array

```
const shuffle = arr => arr.sort(() => Math.random() - 0.5);

// Examples:
console.log(shuffle([1, 2, 3, 4, 5])); // Output: [ 1, 4, 2, 5, 3 ]
console.log(shuffle(['a', 'b', 'c'])); // Output: [ "a", "b", "c" ]
```

This rearranges array elements randomly and is useful for quiz questions, game cards, and featured content.

17. Swap Two Variables

```
[a, b] = [b, a];

// Example:
let a = 5, b = 10;
[a, b] = [b, a];
console.log(a); // Output: 10
console.log(b); // Output: 5
```

This is useful for cleaner logic during sorting or toggling.

18. Flatten an Array